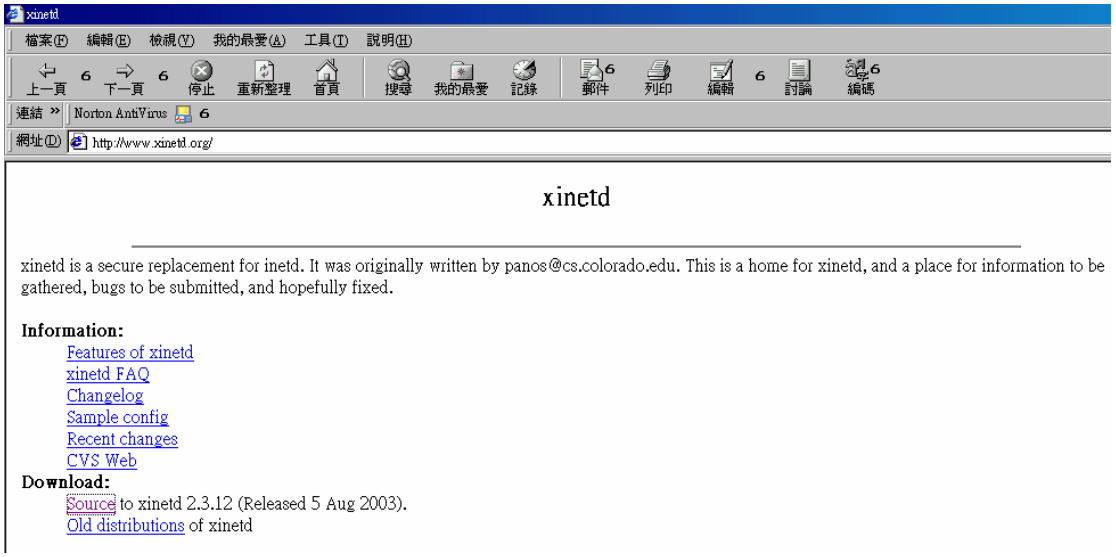
伺服器管理

在資訊發達的現今，使用網站伺服器 Apache、FTP 伺服器、郵件伺服器、NAT、防火牆這些已經越來越重要了。這些在企業網站上的需求也越來越重要。在這一章我們將以伺服器管理為主，如何啟動和關閉我們 Linux 上的伺服器，和如何存取我們這些伺服器。我們在 Linux 中，透過 xinetd 來啟動關閉或重新啟動這些網路伺服器。

下面是安裝 xinetd 網路服務，一般 Linux 系統以經安裝完成，我們可以到 1-1 啟動伺服器。我們可以從 www.xinetd.org 下載 xinetd 網路服務。



我們將下載的原始檔解壓縮。

tar zxvf xinetd-2.3.12.tar.gz

進入 xinetd-2.3.12 的目錄，並且組態 xinetd

cd xinetd-2.3.12

#./configure

組態完後，我們開始編譯 xinetd。

#make

然後我們使用 make install 安裝 xinetd。

#make install

我們可以重新啟動 xinetd 網路服務。

/sbin/service xinetd restart

```
[root@flash xinetd-2.3.12]# /sbin/service xinetd restart
停止 xinetd:[  確定  ]
啓動 xinetd:[  確定  ]
```

如果我們要啟動遠端登錄 telnet **服務**

我們編籍在/etc/xinetd.d/**目錄下的** telnet，**並且將** disable **從** no **改成** yes。

vi /etc/xinetd.d/telnet

```
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
        disable = yes
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
}
```

然後我們再重新啟動 xinetd **網路服務**。

/sbin/service xinetd restart

1-1 **啟動伺服器**

常駐行程 daemon **就是在作業系統中一個與其它程式同步執行的程式。我們的網站伺服器程式就是** Apache **網站伺服器，它執行的程式為** httpd。

```
[root@flash chaiyen]# ps -aux|grep 'httpd'
root      1864  0.0  0.5 19748 2744 ?        S    Sep03   0:01 /usr/sbin/httpd
apache    1893  0.0  0.7 19972 4040 ?        S    Sep03   0:00 [httpd]
apache    1894  0.0  0.8 19960 4440 ?        S    Sep03   0:00 [httpd]
apache    1895  0.0  0.8 19968 4520 ?        S    Sep03   0:00 [httpd]
apache    1896  0.0  0.7 19956 4076 ?        S    Sep03   0:00 [httpd]
apache    1897  0.0  0.8 19968 4528 ?        S    Sep03   0:00 [httpd]
apache    1898  0.0  0.8 19968 4192 ?        S    Sep03   0:00 [httpd]
apache    1899  0.0  0.7 19972 3840 ?        S    Sep03   0:00 [httpd]
apache    1900  0.0  0.7 19880 3908 ?        S    Sep03   0:00 [httpd]
root     20690  0.0  0.1  4448  648 pts/0    S    09:21   0:00 grep httpd
```

在 RedHat Linux Fadora 1 **系統中，我們可以指定啟動和停止我們伺服器。這個程式在**/etc/rc.d/init.d **的目錄。我們可以啟動網站伺服器，使用**/etc/rc.d/init.d/httpd start **這個方式。我們也可以使用**/etc/rc.d/init.d/httpd stop **來關閉它**。

/etc/rc.d/init.d/httpd start

```
[root@flash xinetd.d]# /etc/rc.d/init.d/httpd start
啓動 httpd:[  確定  ]
```

/etc/rc.d/init.d/httpd stop

```
[root@flash chaiyen]# /etc/rc.d/init.d/httpd stop
停止 httpd:
[  確定  ]
```

我們也可以在開機時系統自動啟動伺服器程式。一個自動啟動且持續執行的伺服器稱為 stand-alone 伺服器。當我們啟動作業系統時，RedHat Linux Fadora 1 使用 SysV 啟始行程來自動啟動伺服器。這個程序使用特別的起始程式來啟動位於 /etc/rc.d/init.d 目錄的伺服器。大部份的 Linux 網站伺服器為自動起動且持續執行，這個程式為 httpd，而且它也是在/etc/rc.d/init.d 的目錄下。在 RedHat Linux Fadora 1 我們可以使用/etc/rc.d/init.d/xinetd restart 程式來重新啟動 xinetd 常駐行程。

#service xinetd restart

我們也可以使用/etc/rc.d/init.d/xinetd restart 來重新啟動 xinetd。

/etc/rc.d/init.d/xinetd restart

```
[root@flash xinetd.d]# /etc/rc.d/init.d/xinetd restart
停止 xinetd:[  確定  ]
啓動 xinetd:[  確定  ]
```

我們也可以使用 xinetd 程式來關閉 xinetd 常駐行程。這會停止 xinetd 所管理 /etc/xinetd.conf 檔案中或/etc/xinetd.d 目錄下的所有伺服器。

#service xinetd stop

/etc/rc.d/init.d/xinetd stop

```
[root@flash xinetd.d]# /etc/rc.d/init.d/xinetd stop
停止 xinetd:[  確定  ]
```

我們也可以使用 xinetd 程式來啟動 xinetd 常駐行程。這會啟動 xinetd 所管理 /etc/xinetd.conf 檔案中或/etc/xinetd.d 目錄下的所有伺服器。

#service xinetd start

/etc/rc.d/init.d/xinetd start

```
[root@flash xinetd.d]# /etc/rc.d/init.d/xinetd start
啓動 xinetd:[  確定  ]
```

1-2 伺服器管理工具

在 RedHat Linux Fadora 1 作業系統中，它提供我們 chkconfig 指令和 redhat-config-services 指令來啟動或關閉我們指定的伺服器。我們可以使用 chkconfig 和 redhat-config-services 來啟動或關閉 Samba 伺服器、Apache 網站伺服器、MySQL 資料庫伺服器、FTP 伺服器、NFS 網路檔案系統伺服器和 Telnet 遠端登錄伺服器。

Chkconfig 和 redhat-config-services 工具管理/etc/rc.d/init.d 目錄下的伺服器服務。服務和啟動層級我們可以指定。執行層級目錄是在/etc/rc.d 的目錄，總共有七個層級，像是/etc/rc.d/rc0.d 到/etc/rc.d/rc6.d。

```
[root@flash chaiyen]# cd /etc/rc.d
[root@flash rc.d]# ls
init.d  rc0.d  rc2.d  rc4.d  rc6.d      rc.sysinit
rc      rc1.d  rc3.d  rc5.d  rc.local
```

1-2-1redhat-config-services

我們也可以介由 redhat-config-services 來啟動我們的服務。Redhat-config-servies 提供一個 Gnome 圖型化界面來簡易使用。我們可以啟動、關閉和重新啟動它。我們也可以設定服務啟動的層級，就像我們使用 chkconfig 指令一樣。我們可以選擇何種服務可以啟動並且選取它的層級，這預設的層級是 5。

```
[root@flash root]# redhat-config-services &
[1] 20826
```

#redhat-config-services &

我們也可以使用 redhat-config-services **來設定服務執行的層級，我們先選取要提供的服務，再選取其編輯執行等級。**

1-2-2chkconfig

我們可以使用 chkconfig **指令來指定服務和希望服務執行的層級。我們可以使用** on **選項來指定服務的執行層級，也可以使用** off **選項來關閉，我們可以使用**—level **選項來執行。**

這 chkconfig **執行可以有**/sbin/chkconfig **或者是輸入** chkconfig **來執行。這是我們啟動** httpd **網站伺服器，並且執行層級為** 5。

/sbin/chkconfig --level 5 httpd on

我們使用 off **選項，當** runlevel **為** 3 **時，則網站伺服器會關閉。**

/sbin/chkconfig --level 3 httpd off

這 reset **選項將儲存服務到** chkconfig **預設選項。**

/sbin/chkconfig httpd reset

我們可以觀看服務的啟動資訊使用—list **選項。**

/sbin/chkconfig --list httpd

```
[root@flash xinetd.d]# /sbin/chkconfig --list httpd
httpd           0:關閉  1:關閉  2:關閉  3:關閉  4:關閉  5:關閉  6:關閉
```

選項	說明
--level 執行層級	設定一個執行層級來關閉、啟動或重設服務。
--list 服務	列出不同執行層級的啟動資訊。所有的服務都會列出，包含 xinetd 服務。
--add 服務	增加服務，建立連接到預設指定的執行層級。
--del 服務	在所有執行層級目錄刪除所有服務的連接。
Service on	啟動服務，在指定或預設的目錄建立啟始連接。
Service off	關閉服務，在指定或預設的目錄建力關閉連接。
Service reset	重設服務啟動資訊，在服務的 init.d 啟動區建立預設連接。

Chkconfig 也能夠有 xinetd 服務的啟動或關閉能力。我們可以進入 xinetd 服務使用 on 或 off 選項。啟動 Samba 組態伺服器 swat，它是在 xinetd 執行，我們可以使用 chkconfig swat on。

/sbin/chkconfig swat on

/sbin/chkconfig --list swat

我們可以在/etc/xinetd.d/目錄下的 swat 將它的 disable 改為 no，這樣我們就可以啟動 Samba 組態伺服器 swat。

#vi /etc/xinetd.d/swat

disable=no

```
# default: off
# description: SWAT is the Samba Web Admin Tool. Use swat \
#              to configure your Samba server. To use SWAT, \
#              connect to port 901 with your favorite web browser.
service swat
{
        disable = yes
        port            = 901
        socket_type     = stream
        wait            = no
        only_from       = 127.0.0.1
        user            = root
        server          = /usr/sbin/swat
        log_on_failure  += USERID
}
```

假如我們要關閉 Samba **組態伺服器，我們可以使用** chkconfig swat off。Off **選項將會將**/etc/xinetd.d/swat **的** disable **設為** yes。

#/sbin/chkconfig swat off

#vi /etc/xinetd.d/swat

```
service swat
{
        disable = yes
        port            = 901
        socket_type     = stream
        wait            = no
        only_from       = 127.0.0.1
        user            = root
        server          = /usr/sbin/swat
        log_on_failure  += USERID
}
```

假如我們希望整個的移除服務，我們可以使用—del **選項**。--del **會移除在執行層級目錄的所有啟動和關閉**。

#/sbin/chkconfig --del httpd

我們可以使用—add **選項重新儲存或加入**。

#/sbin/chkconfig --add httpd

我們可以使用 chkconfig –list **來觀看所有服務的列表**。

/sbin/chkconfig --list

```
[root@flash etc]# /sbin/chkconfig --list|more
microcode_ctl   0:關閉  1:關閉  2:開啓  3:開啓  4:開啓  5:開啓  6:關閉
kudzu           0:關閉  1:關閉  2:關閉  3:開啓  4:開啓  5:開啓  6:關閉
syslog          0:關閉  1:關閉  2:開啓  3:開啓  4:開啓  5:開啓  6:關閉
netfs           0:關閉  1:關閉  2:關閉  3:開啓  4:開啓  5:開啓  6:關閉
network         0:關閉  1:關閉  2:開啓  3:開啓  4:開啓  5:開啓  6:關閉
random          0:關閉  1:關閉  2:開啓  3:開啓  4:開啓  5:開啓  6:關閉
rawdevices      0:關閉  1:關閉  2:關閉  3:開啓  4:開啓  5:開啓  6:關閉
pcmcia          0:關閉  1:關閉  2:開啓  3:開啓  4:開啓  5:開啓  6:關閉
saslauthd       0:關閉  1:關閉  2:關閉  3:關閉  4:關閉  5:關閉  6:關閉
keytable        0:關閉  1:開啓  2:開啓  3:開啓  4:開啓  5:開啓  6:關閉
apmd            0:關閉  1:關閉  2:開啓  3:開啓  4:開啓  5:開啓  6:關閉
atd             0:關閉  1:關閉  2:關閉  3:開啓  4:開啓  5:開啓  6:關閉
gpm             0:關閉  1:關閉  2:開啓  3:開啓  4:開啓  5:開啓  6:關閉
autofs          0:關閉  1:關閉  2:關閉  3:開啓  4:開啓  5:開啓  6:關閉
iptables        0:關閉  1:關閉  2:開啓  3:開啓  4:開啓  5:開啓  6:關閉
irda            0:關閉  1:關閉  2:關閉  3:關閉  4:關閉  5:關閉  6:關閉
isdn            0:關閉  1:關閉  2:開啓  3:開啓  4:開啓  5:開啓  6:關閉
sshd            0:關閉  1:關閉  2:開啓  3:開啓  4:開啓  5:開啓  6:關閉
portmap         0:關閉  1:關閉  2:關閉  3:開啓  4:開啓  5:開啓  6:關閉
nfs             0:關閉  1:關閉  2:關閉  3:關閉  4:關閉  5:開啓  6:關閉
nfslock         0:關閉  1:關閉  2:關閉  3:開啓  4:開啓  5:開啓  6:關閉
sendmail        0:關閉  1:關閉  2:開啓  3:開啓  4:開啓  5:開啓  6:關閉
```

```
以 xinetd 爲主的服務:
        chargen-udp:    關閉
        rsync:  關閉
        chargen:        關閉
        daytime-udp:    關閉
        daytime:        關閉
        echo-udp:       關閉
        echo:   關閉
        services:       開啓
        servers:        開啓
        time-udp:       關閉
        time:   關閉
        dbskkd-cdb:     關閉
        cups-lpd:       關閉
        sgi_fam:        開啓
```

chkconfig 在/etc/rc.d 目錄中建立啟動和關閉適當的執行層級。當 chkconfig 在執行層級 5 增加 httpd 服務，它在/etc/rc.d/rc5.d 目錄建立連接來啟動在/etc/rc.d/init.d 目錄下的 httpd 網站伺服器服務。當我們從執行層級 3 來關閉網站伺服器，chkconfig 會建立一個關閉連接到/etc/rc.d/rc3.d 目錄來使用在/etc/rc.d/initd 目錄下的 httpd，並且關閉這個網路服務。

在這裏，使用者建立在執行層級 3 啟動網站伺服器/etc/rc.d/rc3.d/S85httpd。

```
# ls /etc/rc.d/rc3.d/*httpd
/etc/rc.d/rc3.d/S85httpd
```

我們在執行層級 3 關閉網站伺服器/etc/rc.d/rc3.d/k15httpd。

```
# /sbin/chkconfig --level 3 httpd off
# ls /etc/rc.d/rc3.d/*httpd
/etc/rc.d/rc3.d/K15httpd
```

當我們啟動 httpd 網站伺服器的服務，我們可以使用 chkconfig 來啟動 httpd 在執行層級 3、4 和 5。

```
# /sbin/chkconfig httpd on
```

1-3SysV 起始程式

常駐行程的啟動和關閉程式都放在/etc/rc.d/init.d 的目錄。這些程式通常和該服務伺服器的名稱一樣。 Apache 網站伺服器的程式/usr/sbin/httpd 對應到/etc/rc.d/init.d/httpd。/etc/rc.d/init.d/httpd 真的啟動和關閉網站伺服器。使用 init.d 起始程式來啟動伺服器稱為 SysV 起始程式。/etc/rc.d/init.d 目錄在開機時自動會被啟動。因此放在/etc/rc.d/init.d 目錄裏面的程式就會被啟動。

執行層級	rc.d 目錄	說明
0	rc0.d	關閉
1	rc1.d	單一使用者模式(無網路、限制容量)
2	rc2.d	多人使用者模式(沒有支援 NFS 網路檔案系統)
3	rc3.d	多人使用者模式(全工能模式)
4	rc4.d	沒有使用到
5	rc5.d	多人使用者模式，支援圖型化界面登錄。
6	rc6.d	重新啟動系統。

大部份的伺服器軟體使用 RPM 自動安裝然後在 rc.d 目錄下建立適當的連接。為了自動啟動我們一開始在/etc/rc.d/init.d 的目錄下建立啟動程式，然後在/etc/rc.d/rc3.d 和/etc/rc.d/rc5.d 的目錄下建立符號連接到/etc/rc.d/init.d 的目錄下。我們在/etc/rc.d/init.d/httpd 的檔案中看到不同的選項。而 httpd 網站伺服器的常駐行程都是在/usr/sbin/httpd 執行。

#vi /etc/rc.d/init.d/httpd

```
apachectl=/usr/sbin/apachectl
httpd=${HTTPD-/usr/sbin/httpd}
prog=httpd
RETVAL=0
```

Proccessname 行程名稱為 httpd。

Pidfile：行程編號在/var/run/httpd.pid

Config：網站伺服器的組態檔在/etc/httpd/conf/httpd.conf

```
#!/bin/bash
#
# Startup script for the Apache Web Server
#
# chkconfig: - 85 15
# description: Apache is a World Wide Web server.  It is used to serve \
#              HTML files and CGI.
# processname: httpd
# pidfile: /var/run/httpd.pid
# config: /etc/httpd/conf/httpd.conf

# Source function library.
. /etc/rc.d/init.d/functions

if [ -f /etc/sysconfig/httpd ]; then
        . /etc/sysconfig/httpd
fi
```

這是啟動網站伺服器。daemon 為常駐行程，$httpd 是網站伺服器的路徑/usr/sbin/httpd。

```
start() {
        echo -n $"Starting $prog: "
        check13 || exit 1
        daemon $httpd $OPTIONS
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && touch /var/lock/subsys/httpd
        return $RETVAL
}
```

這是關閉網站伺服器。killproc 為除去網站伺服器的行程。

```
stop() {
        echo -n $"Stopping $prog: "
        killproc $httpd
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && rm -f /var/lock/subsys/httpd /var/run/httpd.pid
}
```

httpd 定義網站伺服器的常駐行程都是在/usr/sbin/httpd 執行。Apachectl 為定義 apache 網站伺服器的執行路徑。

```
# Path to the apachectl script, server binary, and short-form for messages.
apachectl=/usr/sbin/apachectl
httpd=${HTTPD-/usr/sbin/httpd}
prog=httpd
RETVAL=0
```

這是啟動、關閉、狀態和重新啟動的選項。

```
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $httpd
        RETVAL=$?
        ;;
  restart)
        stop
        start
        ;;
  condrestart)
        if [ -f /var/run/httpd.pid ] ; then
                stop
                start
        fi
        ;;
  reload)
        reload
        ;;
  graceful|help|configtest|fullstatus)
        $apachectl $@
        RETVAL=$?
        ;;
  *)
        echo $"Usage: $prog {start|stop|restart|condrestart|reload|status|fullst
atus|graceful|help|configtest}"
        exit 1
esac

exit $RETVAL
```

1-4 延伸網路服務常駐行程(xinetd)

我們有需要 xinetd，當一個遠端的使用者存取這個服務且 xinetd 網路服務行程管理網路服務，當我們的系統接收他們的需求時就會引發 xinetd。xinetd 持續的撿查任何遠端的需求，當它接收需求時，xinetd 會啟動適當的伺服器常駐行程。xinetd 是我們 unix 作業系統上 inetd 的延伸，xinetd 提供安裝和登錄的服務。xinetd 安裝容量可以預防 denial-of-service 的連續攻擊。

我們可以在/etc/rc.d/init.d 目錄中的啟動程式停止 xinetd、啟動 xinetd 和重新啟動 xinetd。

#service xinetd stop

#service xinetd start

#service xinetd restart

在 RedHat Linux Fadora 1 中我們也可以使用 chkconfig 啟動或關閉 xinetd 服務。

#chkconfig swat on

xinetd 的組態檔是/etc/xinetd.conf。

服務是由不同屬性區塊所組成，像是伺服器的名稱、使用協定、和安全限制。每一個網路服務伺服器區塊前都有一個關鍵字 service 和一個獨一無二的伺服器名稱。每一行屬性的開始都會先有屬性名稱然後是分配運算子"="，然後才是數值。一個 default 的關鍵字包含預設的屬性服務。

service <service_name>

{

 <屬性><分配運算子><數值><數值>

}

大部份的屬性都是只有一個數值，我們將數值使用分配運算子分配給屬性。我們也可以使用=+來增加元素，我們也可以使用=-來刪除元素。特定的屬性是服務所必要的。這些屬性包括 socket_type 和 wait.socket_type 可以是 stream，Dgram 可以是以 datagram 為基礎的服務，raw 是直接存取的服務。Wait 可以是 yes(單一執行緒)或 no(多執行緒)。為了標準的網路服務，我們也可以需要提供使用者 ID、伺服器名稱和伺服器使用的協定。Server_args 屬性是讓我們列出想要傳給伺服器名稱的參數。

服務可以被啟動也可以被關閉，我們可以使用 disable 屬性。例如在 telnet 伺服器中，預設是關閉服務，我們可以使用 disable=no 來啟動 telnet 伺服器。然後我們可以重新啟動 xinetd 伺服器來啟動這項服務。

```
#vi /etc/xinetd.d/telnet
service telnet
{
        disable = no
         flags                 = REUSE
         socket_type        = stream
         wait                   = no
         user                   = root
         server                 = /usr/sbin/in.telnetd
         log_on_failure   += USERID
}
```

這是服務常用的屬性與說明。

屬性	**說明**
id	**服務的識別編號。預設的服務** ID **和服務名稱相同。**
type	**服務的類型。**RPC、INTERNAL、UNLISTED。
flags	flags **包含** REUSE、INTERCEPT、NORETRY、IDONLY、NAMEINARGS、NODELAY、DISABLE。
disable	**指定** yes **來取消這個服務的功能。**
socket_type	**指定** stream **來當以** stream **為基礎的服務，** dgram **是以** datagram **為基礎的服務。**
protocol	**指定服務的協定。這協定存在於**/etc/protocols **中。假如這個屬性沒有定義預定為該服務使用的協定。**
wait	**指定這是單執行緒或多執行緒。假如為** yes **則為單一執行緒。假如為** no **則為多執行緒，** xinetd **將持續處理新的需求。**
user	**指定伺服器行程的使用者** ID。**這使用者名稱一定存在於**/etc/passwd。
group	**指令伺服器行程的群組** ID。**這群組名稱存在於**/etc/group。
instances	**指定可以同時被服務啟動的伺服器數量。**
nice	**指定伺服器的優先權。**

屬性	說明
server	指定這個服務的執行程式。
server_args	列出傳給這個伺服器的參數。這不包含伺服器名稱。
only_from	控制可以得到此項服務的遠端主機。這是一串的 IP 位址。假如沒有指定 IP 位址，則拒絕所有位址的存取。
no_acess	拒絕遠端主機的存取。
access_times	指定可以存取服務的時間間距。Hour:min-hour:min。
log_type	記錄檔的類型是 SYSLOG 或是 file 檔案。
log_on_success	當伺服器啟動或關閉時，指定資訊所記錄的地方。這些記錄資訊包含 PID、HOST、USERID、離開狀況和服務時間。
log_on_failure	當伺服器不能啟動時，指定記錄資訊。這資訊包括 HOST、USERID、ATTEMPT 和遠端希望存取的記錄。
rpc_version	指定 RPC 服務的版本。
rpc_number	指定 UNLISTED RPC 服務的編號。
env	定義服務的環境變數。

屬性	說明
passenv	將傳給伺服器的 xinetd 環境變數。
port	指定伺服器的連接埠。
redirect	允許 TCP 服務導向到其它主機。
bind	允許服務限定在指定主機的界面。
interface	允許服務限定在指定主機的界面。
banner	當連接到服務建立時，會顯示遠端主機的檔案名稱。
banner_success	當連接到服務允許時，會顯示遠端主機的檔案名稱。
banner_fail	當連接到服務失敗時，會顯示遠端主機的檔案名稱。
groups	允取存取服務有存取權的群組
enabled	指定服務啟動。
include	插入指定檔案的內容，就像主態檔的一部份
includedir	以”includedir /etc/xinetd.d”來當作目錄。每一個在這目錄裏面的檔案將被循序讀取當作是 xinetd 組態檔，合併組成 xinetd 組態檔。

我們編輯/etc/xinetd.d/tftp 來啟動 tftp 伺服器。

#vi /etc/xinetd.d/tftp

```
# default: off
# description: The tftp server serves files using the trivial file transfer \
#       protocol.  The tftp protocol is often used to boot diskless \
#       workstations, download configuration files to network-aware printers, \
#       and to start the installation process for some operating systems.
service tftp
{
        disable = yes
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
```

這是我們重新啟動 xinetd 服務。

#service xinetd restart

我們在 tftp 伺服器中可以增加有關連接的登錄資訊和伺服器優先權。這 log_onsuccess 將會記錄持續的時間和使用者 ID， log_on_failure 將會記錄使用者連接失敗的記錄，而 nice 會提高優先權到 10。Log_type 指定登錄的資訊記錄的地方，這可以是 SYSLOG 系統記錄檔或是 FILE 指定的檔案。Log_on_success 將指定當連接時要被記錄的資訊 Log_on_failure 將指定當失敗時要被記錄的資訊

log_on_success += DURATION USERID

log_on_failure += USERID

nice = 10

log_type = SYSLOG authpriv

log_on_success = HOST PID

log_on_failure = HOST RECORD

為了安全的考量，我們可以使用 only_from 屬性來限制可以存取的範圍。no_accee 屬性限制列表的主機。instance 屬性限定一次可以連接的伺服器行程數量到 50。

only_from = 61.218.29.2

only_from = localhost

no_access = 61.218.29.6

instances = 50

這是預設的區塊範例。

```
defaults
{
 instances=60
 log_type=FILE /var/log/sesrvicelog
 log_on_success=HOST PID
 log_on_failure = HOST RECORD
 only_from    =    61.218.29.2
 only_from    =    localhost
 no_access    =    61.218.29.6
}
```

這是 xinetd **網路服務預設的屬性。我們可以將** xinetd.conf **組態檔分割成幾個組態檔，每一個組態檔都有它自己的服務。我們可以使用** includedir **屬性將這些分割的組態檔包括起來，我們只要指定要包括的目錄。在** xinetd.conf **組態檔中，我們使用** includedir /etc/xinetd.d **將**/etc/xinetd.d **目錄下的所有檔案給包括起來。**

#vi /etc/xinetd.conf

```
#
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
}

includedir /etc/xinetd.d
```

這是/etc/xinetd.d 目錄下的 swat，這些都會被 includedir 包含到 xinetd.conf 組態檔中。這 disable 屬性的值為 yes，所以不會啟動這個服務。

#vi /etc/xinetd.d/swat

```
# default: off
# description: SWAT is the Samba Web Admin Tool. Use swat \
#              to configure your Samba server. To use SWAT, \
#              connect to port 901 with your favorite web browser.
service swat
{
        disable = yes
        port            = 901
        socket_type     = stream
        wait            = no
        only_from       = 127.0.0.1
        user            = root
        server          = /usr/sbin/swat
        log_on_failure  += USERID
}
```

這是/etc/xinetd.d 目錄下的 telnet，這些都會被 includedir 包含到 xinetd.conf 組態檔中。這 disable 屬性的值為 no，所以會啟動這個服務。

#vi /etc/xinetd.d/telnet

```
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
        disable = no
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
}
```

1-5 使用 TCP Wrappers 來控制存取

保護我們系統最好的方法就是不要安裝各式各樣的服務。我們可以使用 /etc/hosts.allow 和/etc/hosts.deny 來控制主機的存取。我們可以增加下列的一些規則到/etc/hosts.allow 和/etc/hosts.deny 的檔案中。

TCP Wrappers 讀取/etc/hosts.allow 檔案，假如存取是明確的被允許。

TCP Wrappers 會讀取/etc/hosts.deny，假如存取是明確的被拒絕。

hosts.aloow 的檔案包含每個允許存取的主機。我們可以在 hosts.deny 來拒絕所有設定的 IP 存取。我們編輯/etc/hosts.allow 來作下面的設定。

#vi /etc/hosts.allow

我們可以設定所有人皆可以存取網站伺服器 http。

http:ALL

aasir.com 可以存取我們所有的網路資源。

ALL：aasir.com

所有人可以存取我們的 ftp 網路檔案傳輸伺服器。

ftp:ALL

課後練習

1. 在資訊發達的現今，使用網站伺服器 Apache、FTP 伺服器、郵件伺服器、NAT、防火牆這些已經越來越重要了。這些在企業網站上的需求也越來越重要。在這一章我們將以伺服器管理為主，如何啟動和關閉我們 Linux 上的伺服器，和如何存取我們這些伺服器。我們在 Linux 中，透過下列何者來關閉或重新啟動這些網路伺服器？

A.startx

B.init.d

C.telnet

D.xinetd

2. 常駐行程的啟動和關閉程式都放在/etc/rc.d/init.d 的目錄。這些程式通常和該服務伺服器的名稱一樣。Apache 網站伺服器的程式/usr/sbin/httpd 對應到/etc/rc.d/init.d/httpd。

A./etc/rc.d/init.d

B./usr/etc/rc.d/init.d

C./etc/httpd

D./usr/sbin

3. xinetd 持續的撿查任何遠端的需求，當它接收需求時，xinetd 會啟動適當的伺服器常駐行程。xinetd 是我們 unix 作業系統上 inetd 的延伸，xinetd 提供更為安裝和登錄的服務。xinetd 安裝容量可以預防 denial-of-service 的連續攻擊。為了安全的考量，我們可以使用 only_from 屬性來限制可以存取的範圍。請問下列何者屬性限制主機的存取?

A. log_type = SYSLOG authpriv

B. log_on_failure = HOST RECORD

C. log_on_success = HOST PID

D. no_accee = 61.218.29.18

4. hosts.aloow 的檔案包含每個允許存取的主機。我們可以在 hosts.deny 來拒絕所有設定的 IP 存取。請問下列何者可以設定所有人皆可以存取網站伺服器 http?

A. http:ALL

B. ALL：aasir.com

C. ALL:http

D. ftp:ALL

5. **我們可以在**/etc/rc.d/init.d **目錄中的啟動程式停止** xinetd**、啟動** xinetd **和重新啟動** xinetd**。請問下列何者指令錯誤**?

A.#service xinetd stop

B.#service xinetd start

C.#service xinetd restart

D.#/etc/rc.d/init.d start

答案

1.D 2.A 3.D 4.A 5.D